

Patent Application

for

OBJECT MANIPULATORS AND FUNCTIONALITY

Inventor(s):

SCOTT ANTHONY ARVIN
MARC W. SCHINDEWOLF
ROSTISLAW STARODUB
PAUL JOSEPH MCARDLE
MANDAR SHASHIKANT GARGE

Prepared By:

Jason S. Feldmar
Gates & Cooper LLP
Howard Hughes Center
Suite 1050
6701 Center Drive West
Los Angeles, California 90045

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit under 35 U.S.C. Section 119(e) of the following co-pending and commonly-assigned U.S. provisional patent application(s), which is/are incorporated by reference herein:

- 5 [0002] United States Provisional Patent Application Serial No. 60/408,710, entitled "DISPLAY AND USE OF OBJECT MANIPULATORS", by Scott Anthony Arvin, Marc W. Schindewolf, Rostislav Starodub, Dianne Smith Phillips, Mandar Shashikant Garge, Paul Joseph McArdle, and John G. Ford III, filed on September 6, 2002, Attorney Docket No. 30566.249-US-P1.

- 10 [0003] This application is related to the following co-pending and commonly-assigned patent application, which application is incorporated by reference herein:

[0004] United States Patent Application Serial No. XX/YYY,YYY, entitled "TEMPORARY TEXT AND GRAPHIC FEEDBACK FOR OBJECT MANIPULATORS", by Scott Anthony Arvin, filed on the same date herewith,

- 15 Attorney Docket No. 30566.307-US-U1.

BACKGROUND OF THE INVENTION

1. Field of the Invention.

- [0005] The present invention relates generally to computer graphics programs, and
20 in particular, to a method, apparatus, and article of manufacture for using object manipulators in a computer graphics program.

2. Description of the Related Art.

[0006] The use of graphics programs such as Computer Aided Design (CAD) application programs is well known in the art. CAD drawings often have many components (referred to as objects or graphics objects) that together define the drawing. In most computer applications that create and modify graphic objects, an object manipulator is a standard mechanism used to modify the geometric properties of the graphic object. However, the nature and use of object manipulators in the prior art is limited to that of modifying an object's geometric properties. Such limitations are undesirable and reduce the flexibility and usability of computer graphics applications. These problems may be better understood by describing prior art drawing programs, object properties, object manipulators, and property modification techniques.

Drawing Programs

[0007] CAD applications are often used to create drawings used in the architectural, engineering, and construction (AEC) industry. The drawings are often defined by a collection of one or more graphical elements (referred to as objects), such as lines, circles, polylines, text, or dimensions. For example, a collection of various lines may make up a door or window object. CAD programs may treat each object as a single element for creation, manipulation, and modification. Some CAD programs may also provide objects that are special entities with predefined behaviors and display characteristics. Thus, the objects in a CAD program may be object-oriented objects having various methods/behaviors and properties (including display characteristics).

[0008] Information relating to an object may be entered and defined in one or more properties of the object. For example, users may have the capability to enter information about the style, dimensions, location, schedule data, display information and/or other important characteristics of an object.

5

Object Properties

[0009] To edit and maintain an object and/or properties of the object, various methodologies may be available. For example, a properties window may provide a common “one stop” location where a user can manage object properties without
10 having to use individual typed commands. Such a properties window may list the current settings for all object properties, and allow the properties to be viewed alphabetically or by category. To change a property, the property may be selected from the list and a new value may be selected or entered.

15

Object Manipulators

[0010] As described above, an object manipulator is a prior art mechanism used to modify the geometric properties of a graphic object. Other names for object manipulators are handles, or grips. An object manipulator is typically implemented as a glyph or symbol, such as a small square, displayed at some geometrically
20 meaningful point on the object to be manipulated.

[0011] FIG. 1 illustrates a prior art rectangle 100 that is selected for modification, with object manipulators 102 displayed as filled squares located at each vertex.

Activating and dragging one of the object manipulators 102 modifies the position of

the vertex, and thus the width and height properties of the rectangle.

[0012] The typical way to interact with an object manipulator 102 is to place an identifier (e.g., a cursor) controlled by a pointing device (e.g., a computer mouse) over the manipulator 102, activating the manipulator 102 by clicking a physical button on the pointing device, dragging the pointing device to change the position of the manipulator 102, and finally clicking the pointing device again to set the desired value of the object feature being manipulated. However, in the prior art, object manipulators 102 may only be used to alter the geometric properties (e.g., vertex position, shape, etc.) of an object by dragging the manipulator.

Buttons

[0013] To adjust object properties other than geometric properties, the prior art may rely on the use of buttons. As illustrated in FIG. 2, in standard computer user interfaces, a button 200 is a common control used to initiate an action to modify some aspect of the state of the interface. The button 200 displays text (e.g., “SAVE”) or icon graphics to provide information to the user as to the results of activating the button 200. The typical way to interact with a button 200 is to place an identifier (e.g., a cursor) controlled by a pointing device (e.g., a computer mouse) over the button 200, and activating the button 200 by clicking a physical button on the pointing device. Once activated, an action is performed or the state of the interface is modified.

[0014] Accordingly, what is needed is the capability to edit an object’s properties (geometric and otherwise) in an intuitive, easy, and understandable manner.

SUMMARY OF THE INVENTION

[0015] One or more embodiments of the invention provide a method, apparatus, and article of manufacture for modifying a property of an object using object
5 manipulators. The object manipulators may take various forms, glyphs/shapes, and colors to facilitate ease of use and understanding by the user. Additionally, various additional graphic images may be displayed to assist the user in accomplishing a desired task. Further, a property of an object may be constrained in a variety of manners. Such constraints may be reflected in/by object manipulators and/or the use
10 of object manipulators.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] The patent or application file contains at least one drawing executed in color. Copies of this patent or patent application publication with color drawing(s) will be
15 provided by the Office upon request and payment of the necessary fee.

[0017] Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

[0018] FIG. 1 illustrates a prior art rectangle that is selected for modification, with object manipulators displayed as filled squares located at each vertex;

20 [0019] FIG. 2 illustrates a prior art button;

[0020] FIG. 3 is an exemplary hardware environment used to implement one or more embodiments of the invention;

[0021] FIG. 4 is a block diagram that illustrates the components of a graphics

program in accordance with one or more embodiments of the invention;

[0022] FIG. 5 is a block diagram that illustrates the structure of an object list maintained by a 3D database in accordance with one or more embodiments of the invention;

5 [0023] FIGS. 6A and 6B illustrate the use of a button object manipulator/trigger grip in accordance with one or more embodiments of the invention;

[0024] FIG. 7 illustrates an example of some of the shapes, names, and descriptions that may be possible for a trigger grip in accordance with one or more embodiments of the invention;

10 [0025] FIGS. 8A and 8B illustrate the use and display of object manipulator sets in accordance with one or more embodiments of the invention;

[0026] FIGS. 9A and 9B illustrate the use of object manipulators to align a graphic object in accordance with one or more embodiments of the invention;

[0027] FIG. 10 illustrates the use of object manipulators when multiple objects
15 share the same properties in accordance with one or more embodiments of the invention;

[0028] FIG. 11 illustrates examples of the different types of colors, names, and descriptions of the meaning of the different colors that may be used in accordance with one or more embodiments of the invention;

20 [0029] FIGS. 12A-12D illustrate the reorientation and display of object manipulators in accordance with one or more embodiments of the invention;

[0030] FIGS. 13A-13D illustrate the modification of the manipulator function with discrete key presses in accordance with one or more embodiments of the invention;

[0031] FIGS. 14A-14D illustrate the graphic display of object manipulator function state in accordance with one or more embodiments of the invention;

[0032] FIGS. 15A and 15B illustrate the graphic display of a constrain to line grip constraint (FIG. 15A) and a constrain to circle grip constraint (FIG. 15B) in accordance with one or more embodiments of the invention;

[0033] FIG. 16 illustrates a constrain to line ray by values where movement is constrained to a list of valid lengths in accordance with one or more embodiments of the invention;

[0034] FIG. 17 illustrates the display of an error bitmap image in accordance with one or more embodiments of the invention;

[0035] FIGS. 18A and 18B illustrate the use of a particular glyph shape as an object manipulator to align an object in accordance with one or more embodiments of the invention;

[0036] FIG. 19 illustrates a translucent colored object manipulator in accordance with one or more embodiments of the invention;

[0037] FIGS. 20A-20C illustrate the selective display of object manipulators based on proximity in accordance with one or more embodiments of the invention; and

[0038] FIG. 21 is a flow chart illustrating the logical flow for using an object manipulator to modify a property of a graphic object in accordance with one or more embodiments of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0039] In the following description, reference is made to the accompanying drawings which form a part hereof, and which is shown, by way of illustration, several embodiments of the present invention. It is understood that other

5 embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

Hardware Environment

[0040] FIG. 3 is an exemplary hardware environment used to implement one or
10 more embodiments of the invention. Embodiments of the invention are typically implemented using a computer 300, which generally includes a display device 302, data storage devices 304, cursor control devices 306, and other devices. Those skilled in the art will recognize that any combination of the above components, or any number of different components, peripherals, and other devices, may be used with the
15 computer 300.

[0041] One or more embodiments of the invention are implemented by a computer-implemented graphics program 308 (e.g., a computer-aided design [CAD] program), wherein the graphics program 308 is represented by a window displayed on the display device 302. Generally, the graphics program 308 comprises logic and/or data
20 embodied in or readable from a device, media, carrier, or signal, e.g., one or more fixed and/or removable data storage devices 304 connected directly or indirectly to the computer 300, one or more remote devices coupled to the computer 300 via a data communications device, etc.

[0042] Those skilled in the art will recognize that the exemplary environment illustrated in FIG. 3 is not intended to limit the present invention. Indeed, those skilled in the art will recognize that other alternative environments may be used without departing from the scope of the present invention.

5

Software Embodiments

[0043] FIG. 4 is a block diagram that illustrates the components of the graphics program 308 in accordance with one or more embodiments of the invention. There are three main components to the graphics program 308, including: a Graphical User
10 Interface (GUI) 400, an Image Engine (IME) 402, and a DataBase (DB) 404 for storing objects in Drawing (DWG) files 406.

[0044] The Graphical User Interface 400 displays information to the operator and provides the functionality for the operator's interaction with the graphics program 308.

15 [0045] The Image Engine 402 processes the DWG files 406 and delivers the resulting graphics to the monitor 302 for display. In one or more embodiments, the Image Engine 402 provides a complete application programming interface (API) that allows other computer programs to interface to the graphics program 308 as needed.

[0046] The Database 404 is comprised of two separate types of databases: (1) a 3D
20 database 408 known as the "3D world space" that stores 3D information; and (2) one or more 2D databases 410 known as the "2D view ports" that stores 2D information derived from the 3D information.

Object List

[0047] FIG. 5 is a block diagram that illustrates the structure of an object list 500 maintained by the 3D databases 408 in accordance with one or more embodiments of the invention. The object list 500 is usually comprised of a doubly-linked list having
5 a list head 502 and one or more objects 504 interconnected by edges 506, although other structures may be used as well. There may be any number of different object lists 500 maintained by the 3D databases 408. Moreover, an object 504 may be a member of multiple object lists 500 in the 3D databases 408.

Terminology

[0048] As used herein, certain terms may be used frequently. The following provides definitions and descriptions for such terminology. However, the terminology definitions and descriptions provided herein are not intended to limit the
10 scope of the invention and alternative meanings of such terms remain within the
15 scope of the invention.

[0049] A “glyph” is a symbol that conveys information nonverbally. A glyph may have the following properties:

- Shape – the polygonal shape of the glyph that is drawn.
- Color – the color of the shape, both the glyph’s outline color and its fill color.
- 20 • Fill mode – whether or not the polygon is filled with a specified color.
- Orientation – the rotational orientation of the glyph shape, generally whether or not the shape is transformed to align with the object being edited.

[0050] A “grip” is a glyph that has been positioned on an object and has an active

area within which a pointing device will “snap” to the grip’s position. The grip indicates that an action can be performed on the object, and defines the behavior of that action. A custom grip may have the following characteristics:

- Glyph – the shape that is drawn.
- 5 • Position – the position on the object where the glyph is located.
- Size – the active area of the grip, usually defined in pixels, within which the pointing device will snap to the grip’s position.
- Parameter(s) – one or more parameters that can change while using the grip.
- Behavior – what happens when the user selects the grip and drags the pointing
10 device.
- Geometric constraints – cursor position might be constrained along a specified set of points, a line such as a coordinate axis, or a plane.
- Dimensional constraints – value being changed might be constrained to a user specified precision, or to a pre-specified list of acceptable values, such as a list
15 of standard door sizes.
- Dynamic dimensions – ability to display a temporary dimension when the grip is paused over or selected, and allows direct entry to change dimension value and update object.
- Temporary location of the User Coordinate System (UCS), such as when
20 constrained to a plane.
- Key modifiers, such as Shift, Ctrl, or Alt key, to optionally change the behavior of a particular grip.
- Cool grip – Grips are cool when displayed on an object that is not selected.

- Warm grip - A grip is warm when displayed on an object that is selected.
- Hot grip - A grip becomes hot when the grip is selected.

[0051] The standard unconstrained grip that may appear on graphics objects may have a glyph shape that is identical to the grip area in plan view and the glyph scale may be equal to one (1).

Button Object Manipulators / Trigger Grips

[0052] As described above, the prior art relies on object manipulators (e.g., a grip) and or buttons to alter an object or the object's properties. Prior art object manipulators/grips are also referred to as drag grips. To use the drag grip, the user selects the grip, drags the pointing device, selects a new point, and some parameter on the object is updated based on the vector between the original grip position and the selected point.

[0053] One or more embodiments of the invention combines the behavior of multiple prior art controls (i.e., the object manipulator control/grip and button control) to create an object manipulator (also referred to as a trigger grip) that is used as a button. Accordingly, the trigger grip is a new type of grip that changes an object (i.e., modifies a property [or state of an associated parameter] of a graphic object) by activating the grip/manipulator without the need for conducting a drag operation (or conducting any additional interaction from/with the user).

[0054] The trigger grip immediately performs an action on an object when a user selects it, as opposed to the normal drag grip, which changes a dimension property only when a user drags it. Trigger grips may be used to set discrete properties of an

object, such as the swing direction of a door or the justification of a wall. Trigger grips may also be used to change the grip edit mode of an object (i.e., the manner/mode in which a grip may be used to edit an object). For example, selecting an Edit Grid grip in a curtain wall may initiate an in-place editing mode for the curtain wall grid.

[0055] The trigger grip may behave in various manners when selected by a user. In one or more embodiments, trigger grips perform some action or change some property immediately upon selection. Accordingly, the user selects the grip, and the object is changed in some way. Further, the grip does not remain selected. For example, FIG. 6A shows a rectangle 600 with an object manipulator/trigger grip 602 located at the rectangle's center. By activating the manipulator 602, the fill property of the rectangle 600 may be modified, resulting in the image shown in FIG. 6B.

[0056] Trigger grips may take the form of various shapes. FIG. 7 illustrates an example of some of the shapes, names, and descriptions that may be possible for a trigger grip in accordance with one or more embodiments of the invention.

Using Button Object Manipulators to Change Edit State

[0057] In the prior art, multiple grips may be simultaneously displayed. The display of multiple grips simultaneously may be confusing and may crowd the display.

Further, if multiple grips are located near each other, the user may have difficulty selecting a particular grip. In one or more embodiment of the invention, a button object manipulator may be used to change the manipulators/grips on the object being manipulated from one set of manipulators to another.

[0058] For example, in FIG. 8A, a rectangle has a button manipulator 800 at the rectangle's center and has drag manipulators 802 at each vertex. Activating the button manipulator 800 removes the vertex manipulators 802 and displays a new set of manipulators 804 in the middle of each edge, as shown in FIG. 8B.

5

Using Button Object Manipulators to Set Potential Discrete Values

[0059] As described above, prior art grips were limited to altering geometric properties of an object. However, using the grips to alter the geometric properties could potentially be confusing (e.g., when multiple possible geometric properties are
10 available). Further, prior art grips were not and could not be used to modify discrete values for properties of an object.

[0060] In one or more embodiments of the invention, one or more button object manipulators are used to set one or more discrete object property values. This method can be used to display to the user a discrete number of acceptable property values, or
15 to display manipulators in geometrically meaningful locations on the object when the language-based descriptions of the associated properties are potentially geometrically confusing.

[0061] For example, a graphics application might have a rectangle with possible horizontal alignment values of Left, Center, and Right, and possible vertical
20 alignment values of Top, Middle, and Bottom. Further, the rectangle might be drawn using those alignment properties relative to its specified location. Thus, the rectangle could be drawn in different positions by changing its alignment but not its location.

[0062] FIG. 9A shows a rectangle with a middle-center alignment. The middle-

center alignment location is indicated with horizontal line 902 and vertical line 904.

The dark circle 900 is the object manipulator used to indicate the rectangle's current alignment; while the eight light circles are the object manipulators used to indicate the rectangle's potential alignments. Referring to FIG. 9B, by activating the top-left
5 manipulator 906 with a pointing device, the user can change both the vertical and horizontal alignment properties as shown. Accordingly, the object manipulators 906 may be activated to adjust the alignment property to a particular discrete value depending on which object manipulator is selected.

10 Using Object Manipulator Color to Indicate Potential Changes to Other
Objects

[0063] In prior art computer graphics applications object manipulators may be colored for a variety of reasons. For example, the color of a manipulator may be set to indicate something about the feature being manipulated (e.g. on a solid modeling
15 object, vertex manipulators might be one color, while edge manipulators might be another color). Alternatively, the color of the manipulator may indicate the selection state of the object selected. For example, one object in a selection set containing multiple objects might be one color, while all of the other objects might be another color. Such a coloring of a single object in a set may indicate that the single object is
20 considered the primary selection.

[0064] In one or more embodiments of the invention, the color of object manipulators indicates whether or not the object itself, or other, non-selected, objects will be affected by interacting with the object manipulator. For example, light

colored manipulators might indicate that only the object being manipulated will be affected by the interaction, while dark colored manipulators might indicate that not only the selected object, but all other objects with similar properties will be affected by the interaction. The different colors may be useful to warn the user that the consequences of using the object manipulator extend beyond the selected object.

[0065] FIG. 10 illustrates four rectangles 1000A-1000D, the bottom, shaded rectangles 1000C and 1000D are designated as sharing the same properties, while the top, un-shaded rectangles 1000A and 1000B do not share the same properties.

Interacting with a light colored vertex manipulator (i.e., of rectangle 1000A) results in only the selected object 1000A being modified (and not object 1000B). However, interacting with a dark colored vertex manipulator (i.e., of rectangle 1000C) may result in more than the selected object 1000C being modified (i.e., both rectangle 1000C and 1000D are modified).

[0066] In addition, using a color to indicate that only a property of the object itself will be changed is useful when using a button object manipulator to change the editing state of the object.

[0067] FIG. 11 illustrates examples of the different types of colors, names, and descriptions of the meaning of the different colors that may be used in accordance with one or more embodiments of the invention. As illustrated the Object color may be used to indicate that the value being edited by the grip will only affect the selected object. The Object Light color may be used for grips not in a horizontal plane, to help differentiate them from the usually more numerous horizontally oriented grips. The Object Dark color may be used for a particular type of grip (e.g., a Free From Mass

Element Face Grip available in Architectural Desktop™ available from the assignee of the present invention) and/or for faces pointing away from a viewport.

[0068] The Secondary Object color may be used to identify a set of grips on one object that is related to another object being edited. For example, if the anchor of a door in a wall is being edited, the door's grips use the Object grip color, and the grips aligned with the wall use the Secondary Object grip color.

[0069] The Style color may be used to indicate that the value being edited by the grip is a style value, and will affect all objects with the same style as the selected object.

[0070] The Auxiliary color may be used for miscellaneous grips that do not immediately change an object's properties. For example, the auxiliary color may be used for changing the edit state of a grip or changing the current shape being edited. The Auxiliary Dark color may be used for Free Form Mass Element Edit Face grips, for faces pointing away from the viewport.

[0071] The Hover color may indicate when the user has snapped to the grip, but hasn't selected the grip. Thus, the hover color may be used to provide feedback to the user when there are many grips in a small area.

[0072] Lastly, the Selected color may indicate that the grip has been selected.

View Dependent Visibility of Oriented Object Manipulators

[0073] In some prior art graphic applications that create and modify three-dimensional objects, the object manipulators may be oriented so as to align with the object, making it easier to understand what property the manipulator will modify and

how the property/object will behave.

[0074] In one or more embodiments of the invention, object manipulators oriented to the object being manipulated may be reoriented or not visible in certain viewing directions where the orientation of the manipulator might be visually confusing or indistinct. For example, FIG. 12A shows a three-dimensional view of a box, with a triangular object manipulator 1200 to modify the box's height. FIG. 12B shows the same box viewed from a different direction, where the height manipulator 1200 is difficult to see and interact with.

[0075] FIG. 12C shows the same box viewed from the same direction as used in FIG. 12B, where the height manipulator 1200 has been reoriented to still align with the object being manipulated, but is now easier to see and manipulate.

[0076] FIG. 12D shows the same box viewed from the same direction as used in FIG. 12B, where the height manipulator 1200 is not visible. Accordingly, the visibility of an oriented object manipulator 1200 may be dependent on the displayed view. The capability to prevent the display of an object manipulator that is difficult to see and interact with may have various advantages. For example, such a capability could be used on an object with a large number of manipulators so that the number of displayed manipulators is reduced to avoid confusion.

Modifying Object Manipulator Function with Discrete Key Presses

[0077] In the prior art, an object manipulator may have only one function. Further, in some applications, holding down a control key such as the Shift key modifies or constrains the cursor point relative to the manipulator, such as constraining cursor

movement from the manipulator horizontally or vertically. Another prior art approach is to change the set of manipulators on an object by repeatedly clicking on the object with the pointing device, where each click cycles through a small set of manipulators.

However, the prior art fails to provide the ability for a single manipulator to have
5 multiple functions and the ability to easily step through or select the desired function of the single particular manipulator.

[0078] In one or more embodiments of the invention, some object grips have multiple functions. For example, a face grip on a freeform mass element may have the following possible grip modes:

- 10 • Move ortho
- Move
- Move plane
- Pull ortho
- Pull
- 15 • Push ortho

[0079] The function of an object manipulator may be modified after it has been activated when a control key, such as the Ctrl key, is pressed and released (e.g., and not simply held down). Further, an object manipulator may have a default function/action associated with it that may be invoked when the manipulator is
20 activated with/without using the control key. Once activated (or with the first selection of the manipulator), the control key may be used to cycle to the next editing mode. The control key may only need to be activated/depressed once, and then released. Such activation cycles the grip to the next editing mode. After releasing the

control key, the user may start editing the grip/object. An alternate key, such as pressing the control and shift keys simultaneously, may cycle the grip to the previous editing mode. In this regard, there is no need to continue pressing the control key while editing the grip.

- 5 **[0080]** FIGS. 13A-13D illustrate the modification of the manipulator function with discrete key presses in accordance with one or more embodiments of the invention. In FIG. 13A, a manipulator 1300 located on the edge of a rectangle might be used to change the width of the rectangle immediately after it has been activated. After pressing and releasing the identified control key (e.g., the Ctrl key), the same
- 10 manipulator 1300 might then be used to skew the rectangle, as shown in FIG. 13B. After an additional press and release of the control key, the same manipulator 1300 might then be used to stretch and skew the rectangle at the same time, as shown in FIG. 13C. Continual presses eventually cycle through all (or a subset of) the available manipulator 1300 functions.
- 15 **[0081]** One potential advantage of such capability is that multiple functions can be assigned to one manipulator 1300, thereby reducing the number of individual manipulators 1300 displayed on the object and thereby reducing visual complexity.

Cursor Graphics to Display Object Manipulator Function State

- 20 **[0082]** In the prior art, the function associated with a particular object manipulator may be unknown. For example, without actually activating and experimenting with an object manipulator, the user may not know what function will be performed by activating the manipulator.

[0083] In one or more embodiments of the invention, a graphic image may be displayed near the cursor of the pointing device to graphically indicate the manipulator's current function state. Accordingly, if a single object manipulator has multiple functions, which the user is able to cycle through, the user can easily visually
5 identify the manipulator's current function state.

[0084] For example, referring to FIG. 14A, a manipulator 1400 located on the edge of a rectangle may be used to change the width of the rectangle as one function. A graphic image 1402 indicating such functionality may be displayed near the cursor as illustrated in FIG. 14B. If the function of the same manipulator 1400 was changed in
10 some way to be used to skew the rectangle, a different graphic image 1402 may be displayed, as shown in FIGS. 14C and 14D.

[0085] Such a display of functionality may provide various advantages. For example, by displaying the graphic image, the user may be made aware that different functionality exists for the manipulator. Further, the graphic image may provide some
15 indication of what the current functionality is.

Graphic Display of Object Manipulator Constraints

[0086] When a grip is selected and dragged, the grip's final position may not be the same as the current cursor position, which might be constrained in some way.
20 Accordingly, each grip may have a constraint defined for it, so that user input is constrained to valid values. One of the challenges with using a two-dimensional pointing device in a three-dimensional graphic application is indicating on a two-dimensional viewing screen where the user's current location is in three-dimensional

space. Often the user must visually determine where the current position of the pointing device is in relation to the three-dimensional model the user is working in. This problem is exacerbated when the movement of the pointing device is projected in some way.

- 5 **[0087]** Accordingly, in one or more embodiments of the invention, graphic feedback may be provided to indicate where the current cursor position is located in three-dimensional space and how the current cursor position is constrained. To indicate a position of a cursor and how user input is constrained, a line or arc (e.g., a dotted line or arc) may be drawn from the current cursor position to a constrained point. Such a
- 10 line or arc may be referred to as a projection. Further, when interaction with an object manipulator is constrained in some way, and the current position of a pointing device in three-dimensional space is projected in some way to a point other than the current point, graphics may be drawn between the current point and the resulting point to indicate the progress of that projection.
- 15 **[0088]** Various general constraint types may be used by a grip and may typically work in three-dimensional as well as two-dimensional space. For example, a constrain-to-plane grip constraint may project user input to a point (e.g., the nearest point) on a specified plane. In this regard, if a constraint has a plane defined for it, a point not on the plane may first be constrained to the nearest point on the plane, then
- 20 constrained to some other point or specific geometry depending on the rules of the individual constraint. FIGS. 15A and 15B illustrate the graphic display of a constrain-to-line grip constraint (FIG. 15A) and a constrain-to-circle grip constraint (FIG. 15B).

[0089] Referring to FIG. 15A, a constrain-to-line grip constraint may project user input from the cursor position 1502 to a point 1506 (e.g., the nearest point) on a specified infinite line. If a plane normal is specified along with the line, user input may first be projected from the cursor position 1502 to a point 1504 on the plane (i.e., the nearest point 1504 on the plane), and may then be projected to a point 1506 on the line (i.e., the nearest point on the line).

[0090] Referring to FIG. 15B, a constrain-to-circle grip constraint may project user input from a cursor position 1502 to a point 1506 (e.g., the nearest point) on a specified circle. If a plane is specified, the cursor position 1502 is first projected to a point 1504 on the plane of the circle, and then to a point 1506 on the circle.

[0091] Another example of a type of constraint is a constrain-to-line-ray grip constraint which may project user input perpendicularly to a point on a specified line ray. Similar to the line grip constraint, if a plane normal is specified along with the line ray, user input may first be projected to the plane, and then to the line ray. A constrain-to-radial-ray grip constraint may project user input radially to a point on a specified line ray on a specified plane. In this regard, the point be first be projected to the nearest point on the plane, and then projected to a point on the line ray with the same distance to the ray base point as the point on the plane.

[0092] A constrain-to-line-segment grip constraint may project user input to a point on a specified line segment. However, if the nearest point is not on the line segment, the point may be constrained to the nearest end point.

[0093] Another type of constraint is a constrain-to-polyline grip constraint that may project user input to a point (e.g., the nearest point) on a polyline.

Constraining Object Manipulator Movement to Discrete Dimensional Values

[0094] As described above, various types of constraints may be available. In this regard, an object manipulator that is used to set a dimensional property on an object
5 can be positioned at any point on a plane or in three-dimensional space.

[0095] In one or more embodiments of the invention, the user may have the capability to constrain grip movement to discrete dimensional values. For example, the movement of a grip may be constrained to a pre-specified set of values.

[0096] FIG. 16 illustrates a constrain-to-line-ray-by-values constraint where
10 movement is constrained to a list of valid lengths. As illustrated, the constrain-to-line-ray-by-values grip constraint may project user input from a cursor position 1602 to a point on a specified line ray 1606 and then constrains that point 1606 to the nearest length 1608 from a specified list of valid lengths. If a plane normal is specified along with the line ray, user input may first be projected from the cursor
15 position 1602 to the plane 1604, then to a point 1606 on the line ray, then to the nearest valid value 1608. Thus, the current cursor position 1602 may be projected to the nearest point on the plane 1604, then the nearest point 1606 on the line, then projected to the nearest valid value 1608. As illustrated, to aid in visibility to the user, constraint projection lines may be drawn between points 1602, 1604, 1606A, and 1608 – and not point 1606.
20

[0097] A constrain-to-line-segment-by-values grip constraint is basically a combination of constrain-to-line-ray-by-values constraint and constrain-to-line-segment grip constraint. Typically, a grip with a “by values” constraint may also

provide the user with a means to enter custom values, such as by holding the Ctrl key while dragging. The “by values” part of a constrain-to-line-segment-by-values constraint acts like a constrain-to-line-ray-by-values constraint, in that the valid values may be measured from a single base point. The “custom” part of the constraint
5 acts like a constrain-to-line-segment constraint.

Cursor Graphics to Indicate Constraint Errors

[0098] While graphics may illustrate the various projections for the various types of constraints, the invention may also provide for displaying a bitmap image at the
10 current cursor position if selecting that position will result in an error condition when interacting with an activated object manipulator.

[0099] For example, in FIG. 17 cursor 1702 movement is constrained to a specified line segment whose endpoints 1704 are indicated with the black dots. If the current cursor 1702 position projects to an invalid point 1704, a bitmap image 1706 is
15 displayed at the cursor 1702, as shown at the cursor positions 1702 at the far left and far right.

[0100] In addition to the above, when a constraint has a valid range (e.g., as a line ray or line segment constraint), if the normal constrain results in the constrained point being located beyond the valid range, the linetype for the normal projection graphics
20 may change to a red dashed line.

Object Manipulator Glyph to Indicate Alignment

[0101] Object manipulators may be located at geometric points on the object to be

manipulated. In accordance with one or more embodiments of the invention, the shape of the object manipulator is used to indicate the desired alignment. For example, FIG. 18A illustrates the use of a diamond shape as an object manipulator. The various positions of the diamond indicate alignment with a line, circle, or other shape. The diamond shaped object manipulator can be located at any point along the line or circle and still provide the same function. As illustrated, the diamond is oriented horizontally and may indicate left, center, or right justifications.

Alternatively, when placed on a line, the diamond may indicate justification with the line where the diamond is located (e.g., the baseline of FIG. 18A).

[0102] Referring to FIG. 18B, in a case where alignment along two different axes is required, a cross shaped manipulator 1802 may be used to indicate double alignment, in other words a point. Thus, the shape/glyph of the grip may change to indicate the alignment.

Translucent Object Manipulators

[0103] In the prior art, object manipulators/grips are opaque, which results in parts of the object being manipulated being hidden. Further, object manipulators may also be displayed at a constant size, so if the object is very small in the current view, the entire object may often be hidden behind the opaque manipulators.

[0104] In one or more embodiments of the invention, object manipulators may be displayed with translucent/transparent colors. Such coloring provides the ability to see the object being manipulated behind the manipulators. FIG. 19 illustrates a translucent colored object manipulator 1902 in accordance with one or more

embodiments of the invention. Object manipulators 1902 are translucent so that the door object being manipulated remains visible through the manipulators 1902.

Proximity Dependent Visibility of Object Manipulators

5 [0105] Multiple object manipulators may be defined for the manipulation of a particular object. In the prior art, all of the manipulators are displayed regardless of the current viewing direction of the object. Often, particularly when the object is visually small on the viewing device, two manipulators will overlap each other. Such an overlap of manipulators results in potentially difficult or confusing use or
10 interaction with the manipulators. For example, when a user desires to use a single particular object manipulator on an object with multiple manipulators located confusingly proximate to each other, the user may not know which manipulator the cursor is located over and may not select the desired manipulator.

[0106] One or more embodiments of the invention provide a method of controlling
15 the visibility of an object manipulator based on the manipulator's visual proximity to other manipulators present on the object to be manipulated. All manipulators that can be potentially displayed on an object may be identified with a proximity priority. When two manipulators are displayed within a specified proximity, such as the apparent visual size or the active region of the manipulator, the manipulator with the
20 lower priority may not be displayed, thus simplifying the visual display of the manipulators, and by extension simplifying the use and interaction of the manipulators. The active region of a manipulator may be defined as that region relative to the manipulator graphics that activates interaction with the manipulator

when the user positions the cursor of a pointing device over the region and activates the pointing device.

[0107] For example, FIG. 20A illustrates a rectangle selected in one view, with object manipulators 2002 displayed on the rectangle's vertices and edges. FIG. 20B shows the same rectangle selected in a different view, which has a different zoom factor, making the rectangle appear visually smaller than it is in the view in FIG. 20A. Since object manipulators are usually displayed with a constant relative size and not scaled with the object being manipulated, the edge manipulators on each side overlap with their corresponding vertex manipulators. If the edge manipulators have a lower proximity priority, and they overlap with other manipulators, they are not displayed, as shown in FIG. 20C.

Visibility of Object Manipulators based on Selection Quantity

[0108] As described above, once a number of object manipulators have been defined for a particular object, those manipulators are always displayed on the object when it is selected, regardless of the apparent visual scale of the object and the number of other objects currently selected. Such displaying of all object manipulators often results in an extremely large number of manipulators displayed at the same time, occluding the objects that are selected. In addition, when a large number of objects are selected the user will rarely, if ever, be interested in manipulating single properties of individual objects, but is more interested in accomplishing general operations on all of the selected objects, e.g. moving, copying, or deleting them.

[0109] In one or more embodiments of the invention whether or not manipulators

are displayed on an object is based on the number of objects selected the first time. If there are currently no objects selected for editing, and the number of objects selected is less than a pre-specified number, object manipulators are displayed, otherwise the object manipulators are not displayed.

- 5 **[0110]** For example, suppose the user (or default) specifies a value of two or more selected objects. With such a value, when the user selects one object, all of the object manipulators (or a subset thereof) may be displayed. However, when the user selects additional objects or adds another object to the selected set of objects, the object manipulators are no longer displayed. The lack of displayed manipulators assumes
- 10 that the user intends to control the objects as a group and does not desire to edit or manipulate the individual objects in the set. The lack of displayed manipulators also minimizes the display to avoid confusing the user.

Logical Flow

- 15 **[0111]** As described above, object manipulators (both button and traditional) provide the ability to easily modify one or more properties of an object. FIG. 21 is a flow chart illustrating the logical flow for using an object manipulator to display an object manipulator (button or traditional) and modify a property of a graphic object in accordance with one or more embodiments of the invention.

- 20 **[0112]** At step 2100, the graphic object is displayed in a computer graphics program. The graphic object has one or more properties that may control how the graphic object is displayed. For example, properties may establish the color, size, shape, orientation, alignment, etc. of the graphic object.

[0113] At step 2102, an object manipulator is displayed on the graphic object.

While some embodiments of the invention may utilize button object manipulators, other embodiments may use traditional object manipulators (e.g., that are dragged to edit a property of an object).

- 5 [0114] In some embodiments, a set of two or more object manipulators may be displayed on the graphic object. Each object manipulator in the set may be used to manipulate the graphic object. Further, the user may easily change the set of object manipulators that are displayed. For example, the user may simply press and release the control key to change the set of object manipulators that are displayed and
- 10 currently available to the user. Alternatively, the user may use a button object manipulator displayed on the graphic object to change the set of object manipulators. For example, by selecting and activating (e.g., using a cursor control device) the button object manipulator, the currently available manipulator set may be changed. Thus, the user may easily cycle through the available sets of object manipulators.
- 15 Further, the user may also have the ability to create new sets, and edit or delete existing sets of the object manipulators as desired.

- [0115] In alternative embodiments, multiple button object manipulators may be displayed on the graphic object in geometrically meaningful locations. For example, as described above, when language-based descriptions of multiple properties that are
- 20 associated with multiple button object manipulators are potentially geometrically confusing, the button object manipulators may be located geometrically in a manner to clarify the manipulator's function. For example, the various potential alignment options for the graphic object may be represented by placing various button object

manipulators in strategic geometric locations on the graphic object that reflect those alignment options.

[0116] In addition to the above, object manipulators may be displayed in one or more colors. Such coloring may indicate whether use of the object manipulator to manipulate the object will affect one or more properties of another object. In yet another embodiment, an object manipulator may be reoriented when an initial orientation is visually confusing or indistinct. Such reorientation may make the object manipulator easier to discern. Alternatively, if the reoriented manipulator is visually confusing or indistinct, the manipulator may not be visible at all.

10 [0117] Object manipulators (both button and traditional) may also be displayed in a variety of different forms or glyphs. Different forms or glyphs may also indicate different functionality. For example, a particular glyph shape may indicate that the graphic object is to be aligned with respect to one or more additional objects. In such an example, a diamond shaped glyph may indicate alignment with a line, circle, or
15 other shape object. The glyph may also be displayed in a translucent color so that the graphic object is visible behind an object manipulator.

[0118] At step 2104, the object manipulator is activated. The activation may be invoked by placing a cursor (controlled by a pointing device) over the object manipulator and clicking a button on the pointing device to activate the object
20 manipulator. As described above, the same object manipulator may be used to perform various discrete functions that modify one or more properties of the graphic object. The function that will be performed by a particular object manipulator is represented by the manipulator's function state. Thus, to change the function

performed, after activating the object manipulator, the user may modify the object's function state (e.g., by pressing and releasing a control key). Further, so that the user may more easily identify the current function state, a graphic image may be displayed (e.g., at the location of the cursor) that graphically indicates the function state of the

5 object manipulator

[0119] While object manipulator may often be activated at any time and at a variety of cursor locations, object manipulators may also be constrained to a certain set of values or actions. In this regard, the user may not be permitted to select certain positions when using an object manipulator. Accordingly, after activating an object
10 manipulator, a bitmap image (e.g., a circle with a line through it) may be displayed at the cursor position if selecting the cursor position will result in an error condition when interacting with the manipulator.

[0120] At step 2106, the property of the graphic object is modified in response to the activation. With traditional object manipulators, the manipulator may need to be
15 dragged to modify a property of an object. However, if the object manipulator is a button object manipulator, there is no need to drag the button object manipulator to modify the property. In other words, the property is directly modified simply by activating (e.g., selecting with a cursor and pressing a physical button on a cursor control device) the button object manipulator. Alternatively, once the object
20 manipulator (button or traditional) is activated, a discrete number of acceptable property values may be displayed for the user's selection.

Conclusion

[0121] This concludes the description of the preferred embodiment of the invention.

The following describes some alternative embodiments for accomplishing the present invention. For example, any type of computer, such as a mainframe, minicomputer,
5 or personal computer, or computer configuration, such as a timesharing mainframe, local area network, or standalone personal computer, could be used with the present invention.

[0122] The foregoing description of the preferred embodiment of the invention has been presented for the purposes of illustration and description. It is not intended to be
10 exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.